

FIG. 1

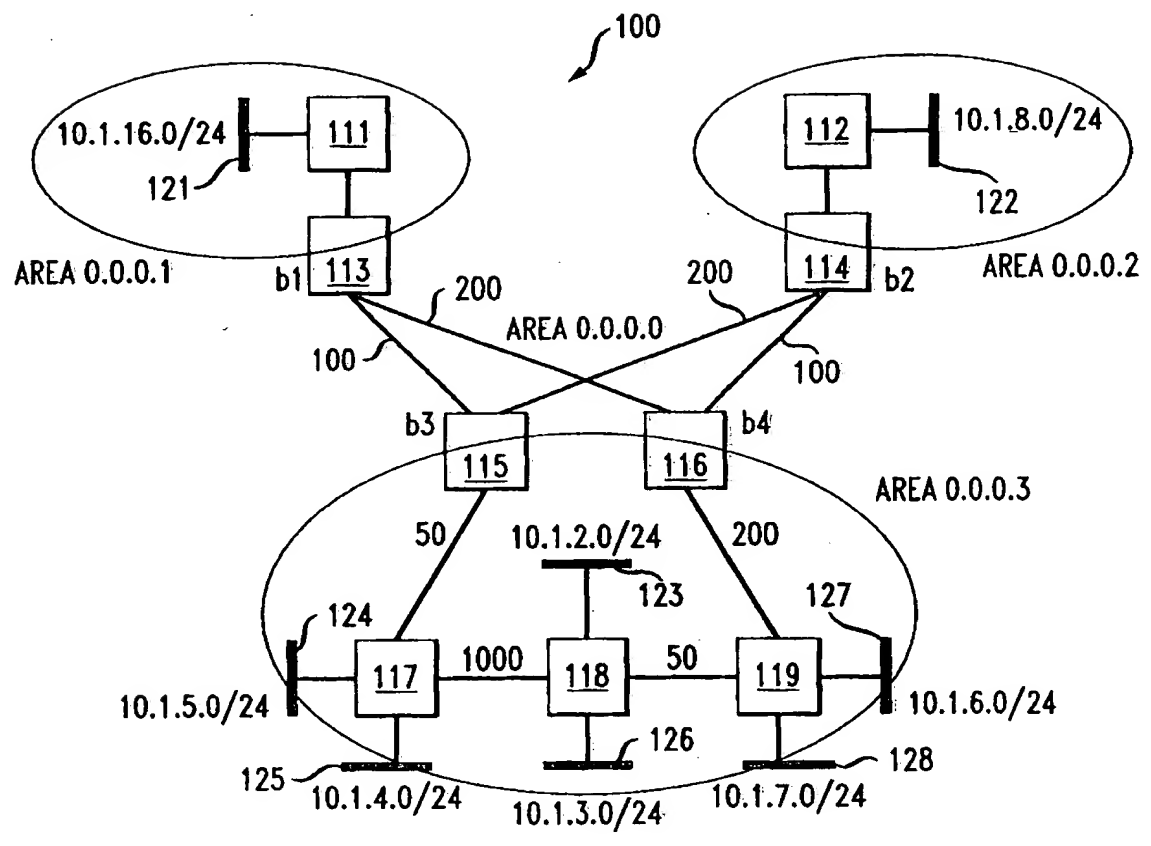


FIG. 2

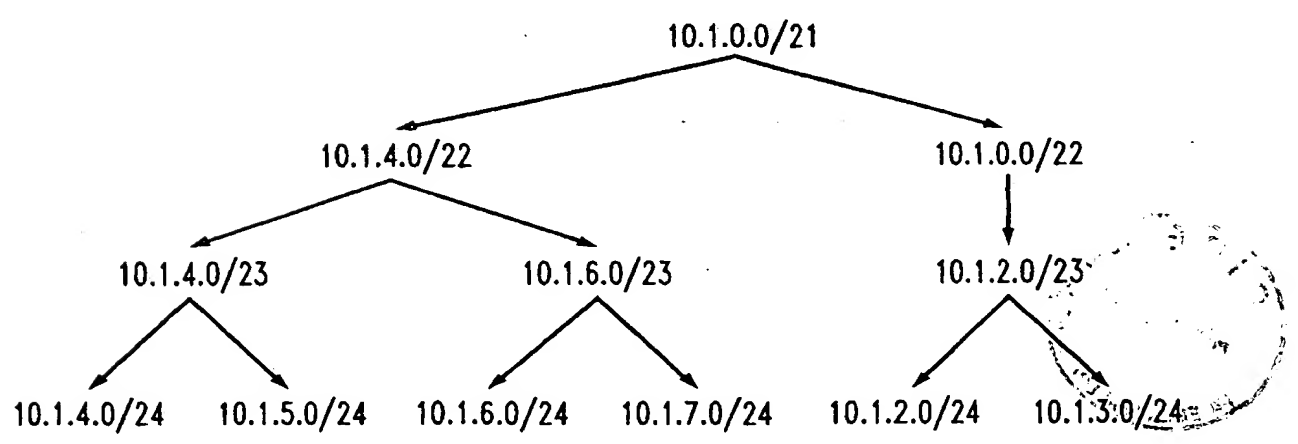


FIG. 3

```

procedure COMPUTEMINERROR(Aggregate  $x$ , Aggregate  $y$ , integer  $l$ )
1. if subTree[ $x$ ,  $y$ ,  $l$ ].computed = true
2.   return [subTree[ $x$ ,  $y$ ,  $l$ ].error, subTree[ $x$ ,  $y$ ,  $l$ ].aggregates]
3.   minError := minError1 := minError2 :=  $\infty$ 
4.   if  $x$  is a leaf {
5.     minError1 :=  $\sum_{s \in S} D(s, t) * (lsp(s, x, \{y\}, W_A) - lsp(s, x))$ 
6.     if  $l > 0$ 
7.       minError2 :=  $\sum_{s \in S} D(s, t) * (lsp(s, x, \{x\}, W_A) - lsp(s, x))$ 
8.     if minError1  $\leq$  minError2
9.       [subTree[ $x$ ,  $y$ ,  $l$ ].error, subTree[ $x$ ,  $y$ ,  $l$ ].aggregates] := [minError1,  $\emptyset$ ]
10.    else
11.      [subTree[ $x$ ,  $y$ ,  $l$ ].error, subTree[ $x$ ,  $y$ ,  $l$ ].aggregates] := [minError2,  $\{x\}$ ]
12.  }
13. if  $x$  has a single child  $u$  {
14.   [minError1, aggregates1] := COMPUTEMINERROR( $u$ ,  $y$ ,  $l$ )
15.   if  $l > 0$ 
16.     [minError2, aggregates2] := COMPUTEMINERROR( $u$ ,  $x$ ,  $l - 1$ )
17.   if minError1  $\leq$  minError2
18.     [subTree[ $x$ ,  $y$ ,  $l$ ].error, subTree[ $x$ ,  $y$ ,  $l$ ].aggregates] := [minError1, aggregates1]
19.   else
20.     [subTree[ $x$ ,  $y$ ,  $l$ ].error, subTree[ $x$ ,  $y$ ,  $l$ ].aggregates] := [minError2, aggregates2  $\cup$   $\{x\}$ ]
21. }
22. if  $x$  has children  $u$  and  $v$  {
23.   for  $i := 0$  to  $l$  {
24.     [minError1, aggregates1] := COMPUTEMINERROR( $u$ ,  $y$ ,  $i$ )
25.     [minError2, aggregates2] := COMPUTEMINERROR( $v$ ,  $y$ ,  $k - i$ )
26.     if minError1 + minError2  $<$  minError
27.       minError := minError1 + minError2
28.       aggregates := aggregates1  $\cup$  aggregates2
29.   }
30.   for  $i := 0$  to  $l - 1$  {
31.     [minError1, aggregates1] := COMPUTEMINERROR( $u$ ,  $x$ ,  $i$ )
32.     [minError2, aggregates2] := COMPUTEMINERROR( $v$ ,  $x$ ,  $k - i - 1$ )
33.     if minError1 + minError2  $<$  minError
34.       minError := minError1 + minError2
35.       aggregates := aggregates1  $\cup$  aggregates2  $\cup$   $\{x\}$ 
36.   }
37.   [subTree[ $x$ ,  $y$ ,  $l$ ].error, subTree[ $x$ ,  $y$ ,  $l$ ].aggregates] := [minError, aggregates]
38. }
39. subTree[ $x$ ,  $y$ ,  $l$ ].computed := true
40. return [subTree[ $x$ ,  $y$ ,  $l$ ].error, subTree[ $x$ ,  $y$ ,  $l$ ].aggregates]

```

09775329.051404

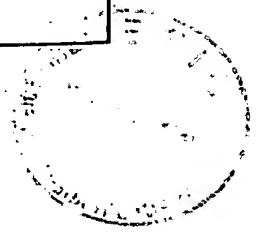


FIG. 4

procedure COMBINEMINERROR()

```

1. for  $i = 1$  to  $m$ 
2.   for  $j = 0$  to  $k$  {
3.      $T_i[j].[\text{error}, \text{aggregates}] := \text{COMPUTEMINERROR}(r(T_i), \epsilon, j)$ 
4.      $X_i[j].[\text{error}, \text{aggregates}] := [\infty, \emptyset]$ 
5.   }
6. for  $j = 0$  to  $k$ 
7.    $X_1[j].[\text{error}, \text{aggregates}] := T_1[j].[\text{error}, \text{aggregates}]$ 
8. for  $i = 1$  to  $m$ 
9.   for  $j = 0$  to  $k$ 
10.    for  $l = 0$  to  $j$ 
11.      if  $(X_{i-1}[l].\text{error} + T_i[j-l].\text{error} < X_i[j].\text{error})$  {
12.         $X_i[j].\text{error} = X_{i-1}[l].\text{error} + T_i[j-l].\text{error}$ 
13.         $X_i[j].\text{aggregates} = X_{i-1}[l].\text{aggregates} \cup T_i[j-l].\text{aggregates}$ 
14.      }

```

FIG. 5

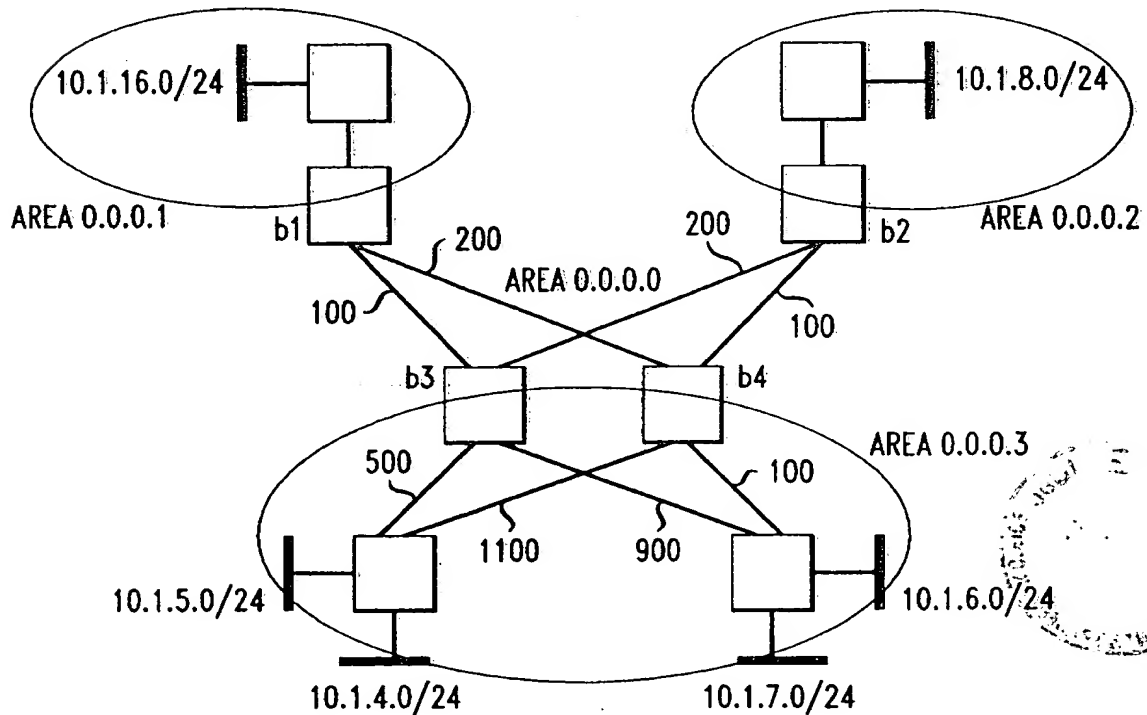


FIG. 6

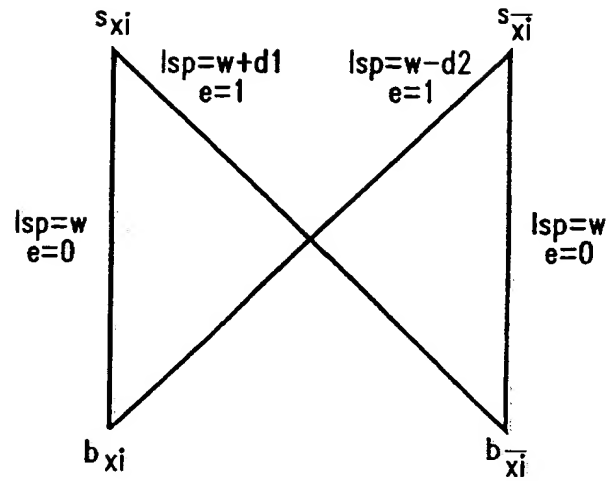


FIG. 7A

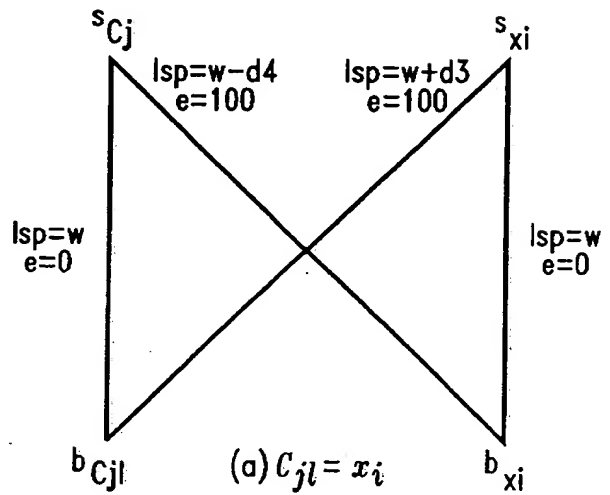
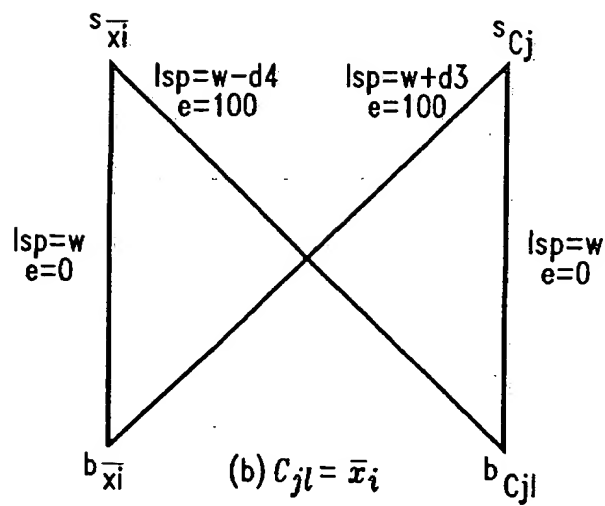


FIG. 7B



09775329.051404

FIG. 8

```

procedure COMPUTEWEIGHTSCUMULATIVE ()
1. for each  $b \in B_i$  set  $W_{min}(b) := 0$ 
2. for  $i := 1$  to  $r$  {
3.    $W := W_{min}$ 
4.   Choose a random subset  $R \subseteq B_i$  of ABRs
5.   for each  $b \in R$  set  $W(b)$  to a random weight in  $[0, L]$ 
6.   if  $\sum_{s \in se(s, B(s, W))} < \sum_{s \in se(s, B(s, W_{min}))}$ 
7.      $W_{min} := W$ 
8. }
9. return  $W_{min}$ 

```

FIG. 9

```

procedure ComputeWeightsMax(Q)
1. for each  $b \in B_i$  set  $Wold(b) := 0$ 
2. while  $(Pb_2B$ 

i  $Wold(b) \leq ($ 

j  $B_i j * (j B_i j - 1)$ 

2 ) *  $lspmax$ ) do 3. Let

Q0 be a new set of inequalities that result when the value  $Wold(b)$  is
substituted for each variable  $W(b)$  only on the LHS of each inequality in
Q 4. Set  $Wnew(b)$  to the smallest possible value such that each
inequality in Q0 is satisfied when  $Wnew(b)$  is substituted for variable  $W$ 
(b) in Q0 5. if  $Wnew = Wold$  6. return  $Wnew$  7. else 8.  $Wold := Wnew$ 
9. g 10. return "there does not exist a weight assignment  $W$ "

```

FIG. 10

```

procedure COMPUTEWEIGHTSTWOABR()
1. Set  $V_{opt} := v(s_1)$ ,  $E := E_{opt} := \sum_{s \in se(s, b_1)}$ 
2. for  $j := 1$  to  $n$  {
3.    $E := E + e(s_j, b_2) - e(s_j, b_1)$ 
4.   if  $E < E_{opt}$ 
5.      $V_{opt} := v(s_{j+1})$ ,  $E_{opt} := E$ 
6. }
7. return  $V_{opt}$ 

```

0975329 0001
 1075329 0001
 1175329 0001
 1275329 0001
 1375329 0001
 1475329 0001
 1575329 0001
 1675329 0001
 1775329 0001
 1875329 0001
 1975329 0001
 2075329 0001
 2175329 0001
 2275329 0001
 2375329 0001
 2475329 0001
 2575329 0001
 2675329 0001
 2775329 0001
 2875329 0001
 2975329 0001
 3075329 0001
 3175329 0001
 3275329 0001
 3375329 0001
 3475329 0001
 3575329 0001
 3675329 0001
 3775329 0001
 3875329 0001
 3975329 0001
 4075329 0001
 4175329 0001
 4275329 0001
 4375329 0001
 4475329 0001
 4575329 0001
 4675329 0001
 4775329 0001
 4875329 0001
 4975329 0001
 5075329 0001
 5175329 0001
 5275329 0001
 5375329 0001
 5475329 0001
 5575329 0001
 5675329 0001
 5775329 0001
 5875329 0001
 5975329 0001
 6075329 0001
 6175329 0001
 6275329 0001
 6375329 0001
 6475329 0001
 6575329 0001
 6675329 0001
 6775329 0001
 6875329 0001
 6975329 0001
 7075329 0001
 7175329 0001
 7275329 0001
 7375329 0001
 7475329 0001
 7575329 0001
 7675329 0001
 7775329 0001
 7875329 0001
 7975329 0001
 8075329 0001
 8175329 0001
 8275329 0001
 8375329 0001
 8475329 0001
 8575329 0001
 8675329 0001
 8775329 0001
 8875329 0001
 8975329 0001
 9075329 0001
 9175329 0001
 9275329 0001
 9375329 0001
 9475329 0001
 9575329 0001
 9675329 0001
 9775329 0001
 9875329 0001
 9975329 0001